UC Santa Barbara
UC Santa Barbara Previously Published Works

Title

Search Tracker: Human-Derived Object Tracking in the Wild Through Large-Scale Search and Retrieval

Authors

Bency, Archith John
Karthikeyan, S
De Leo, Carter
et al.

Search Tracker: Human-Derived Object Tracking in the Wild Through Large-Scale Search and Retrieval

Archith John Bency, *Student Member, IEEE*, S. Karthikeyan, *Student Member, IEEE*, Carter De Leo, Santhoshkumar Sunderrajan, *Member, IEEE*, and B. S. Manjunath, *Fellow, IEEE*

Abstract—Humans use context and scene knowledge to easily localize moving objects in conditions of complex illumination changes, scene clutter, and occlusions. In this paper, we present a method to leverage human knowledge in the form of annotated video libraries in a novel search and retrieval-based setting to track objects in unseen video sequences. For every video sequence, a document that represents motion information is generated. Documents of the unseen video are queried against the library at multiple scales to find videos with similar motion characteristics. This provides us with coarse localization of objects in the unseen video. We further adapt these retrieved object locations to the new video using an efficient warping scheme. The proposed method is validated on in-the-wild video surveillance data sets where we outperform state-of-the-art appearance-based trackers. We also introduce a new challenging data set with complex object appearance changes.

Index Terms—Data-driven methods, video search and retrieval, visual object tracking.

I. INTRODUCTION

OBJECT tracking is a well-studied computer vision problem. Tracking algorithms (or trackers) should be robust to large variations of lighting, scene clutter, and handle occlusions while localizing an object across frames. A number of algorithms [14], [37] have approached the problem of tracking by modeling the appearance of objects as they go through illumination, pose, and occlusion changes in image sequences. Motion models are also incorporated in these algorithms to provide a prior for object location in the current

Fig. 1. Two frames (a) and (b) of a sequence with a pedestrian walking from right to left. The red and green boxes represent the tracker's predicted object location and ground-truth, respectively. Typical appearance-based trackers fail on such low visual-quality video sequences.

frame, given the state of the tracker in previous frames. Recent state-of-the-art algorithms have been tested on real-world data sets [6], [8], [11]. These data sets are usually of good image quality and capture sufficient visual information to distinguish between the object of interest and its surroundings. While tracking objects in videos with low-quality imaging, these methods have difficulty in learning robust appearance and motion models. As video infrastructures like surveillance networks have been around for a decade, it is still important to be able to detect and track objects in legacy low-resolution and low-quality videos.

An example of tracker failure, where appearance-based features are used, is presented in Fig. 1. The appearance-based tracker gets distracted by background clutter of trees and learns an incorrect appearance model. This leads to tracker failure and the object state is lost. In addition, most of the trackers need either object detectors or manual initialization for the methods to start tracking objects. Object detectors [4], [36] are prone to failure on low-quality images as detectors trained on one data set may not have good detection performance on a different data set. In conditions where one may come across a diverse set of objects (say humans, vehicles, animals, etc.), a large number of detectors would be needed to generate detections for the trackers to be effective.

Humans, on the other hand, find tracking objects in such scenarios to be a relatively easy task. Human-annotated bounding boxes are of higher quality than those generated by tracking algorithms. Humans leverage contextual knowledge of both the scene and typical object motion to effortlessly track objects. Directly replicating human knowledge would involve coming up with complex computational models for tracking. This paper describes a method to leverage data sets of human annotated videos to track moving objects in new

videos, the search tracker (ST). We maintain a library of training videos containing objects annotated with bounding boxes. The training videos are then transformed into representative documents, which are indexed along with the provided bounding boxes. These documents encode motion patterns of annotated objects in the training videos.

For tracking to be applied on a new test video, we generate similar documents from this video. These documents are matched against the library documents to find video segments with similar motion patterns. The assumption is that video segments with similar motion characteristics will have similar object annotations. Finally, object annotations corresponding to the retrieved results are transferred and warped to match the motion in the test video better.

The main contributions of this paper are as follows.

1) We present a method that tackles the problem of tracking objects in the wild using a search and retrieval framework by learning long-term motion patterns from a library of training videos.

2) This approach carries out object tracking without dedicated object detectors or manual initialization and is automated in the true sense.

3) This approach demonstrates an empirically effective way of transferring information learned from one data set to apply onto other data sets of very different visual contents such as viewpoints, types of objects, and so on.

The rest of this paper is organized as follows. Section II presents an overview of related work. Section III provides the details of the proposed method with a focus on the offline library generation and the online test video tracking process. Section IV elaborates on the experiments done to validate our approach, and we present our comments and possible future work and conclusions in Sections V and VI, respectively.

II. RELATED WORK

Object tracking is an active research area in the computer vision community. Surveys of object tracking algorithms are provided in [20], [32], and [41]. A large number of tracking algorithms learn an appearance model from the initial frame and adapt it to information from incoming frames. Tracking results in the current frame are incorporated into the tracking model for subsequent frames. This online paradigm is called tracking by detection [10], [30]. The simplest object trackers within this paradigm have used color histograms [3] and template matching [16]. However, these methods are susceptible to tracking errors, which leads to the tracker model incorporating background clutter and occlusions. Multiple-instance learners [1] and trackers based on structured label Support Vector Machines (SVM) [12] have tackled the problem of sampling the right image patches for online learning. Yi et al. [40] propose a visual tracker that is insensitive to the quality of manual initialization. The tracker takes advantage of motion priors for detected target features from optical flow, thereby handling inaccurate initializations. This method still relies on either a manual initialization or an object detector to initialize the tracker reliably in a close neighborhood of the ground truth to be successful.

In addition, there are methods that learn from annotated data sets in order to create priors that aid appearance-based trackers. Manen et al. [21] have proposed an interesting framework that learns how objects typically move in a scene and uses that knowledge as a prior to guide appearance-based trackers to handle occlusions and scene clutter. This method requires annotations of multiple object tracks in the same scene. In contrast, our method can track objects in scenes totally unrelated to the data set we learn from. Rodriguez et al. [29] use a large database of crowd videos to search and find priors in order to guide a linear Kalman-filter-based tracker. The method requires that the query video has a scene appearance similar to retrieved library videos and that the target's position be manually initialized, which are not required for the proposed approach.

On the front of biologically inspired systems, there are several works that leverage human contextual knowledge for computer vision tasks like action recognition [15], object detection [18], [27], and scene classification [31], [33].

III. SEARCH TRACKER

A. Overview of the Approach

We aim to track objects in unseen videos by finding matches for motion patterns among a library of videos with indexed human-generated annotations. There are two distinct phases in the proposed method. The offline phase operates on a library of training videos with annotated bounding boxes. Training videos are transformed into representative documents, which are indexed along with the provided bounding boxes. The documents encode long-term motion patterns of annotated objects. We use optical flow [34] to represent motion information from videos.

During the second phase, a new test video is accepted for tracking. Documents similar to those created for the training videos are generated. These documents incorporate motion patterns across different scales and spatial locations, which can be matched to those in the training library. This enables the use of smaller training libraries to represent diverse motion patterns. The matching and retrieval process handles detection and tracking of multiple objects in the test videos.

Once matches for test video documents from the training database are found, associated annotation bounding boxes are transferred to the test video. Transferred bounding boxes are warped to improve the match with motion characteristics of tracked objects. We utilize nonmaximal suppression to derive the best bounding boxes from the set of warped bounding boxes. Subsequently, a smoothing step is carried out to regularize the scale of bounding boxes for the detected objects.

To summarize, human-generated annotations are leveraged to track moving objects in challenging scenarios without actual human review of the test video. A high-level block diagram depicting the proposed method is presented in Fig. 2. The library creation process and the proposed query scheme are explained in the following section.

B. Offline Library Creation

1) Training Video Library: The training video library consists of around 20 min of publicly available surveillance
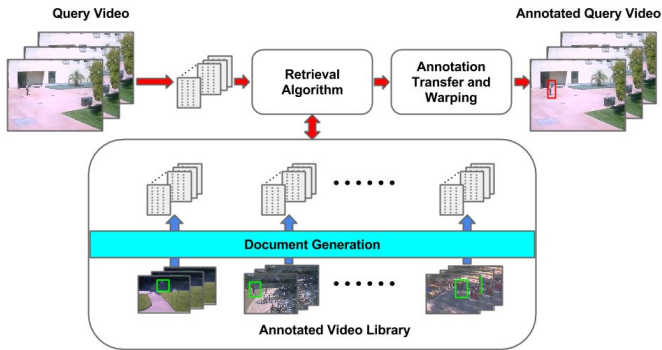
Fig. 2. Block diagram presenting a high-level view of the proposed system. Representative documents are generated from the query video. These documents encode object motion characteristics. Query documents are submitted to the retrieval algorithm to find matches. Annotations corresponding to found matches are then transferred and warped onto the query video. The red arrows represent online steps and the blue arrows represent offline steps of the approach. (Best viewed in color.)
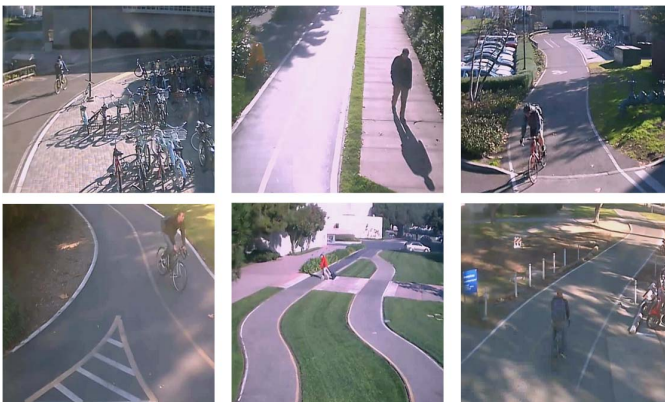

Fig. 3. Example frames from six of the videos included in the library of training videos. (Best viewed in color.)
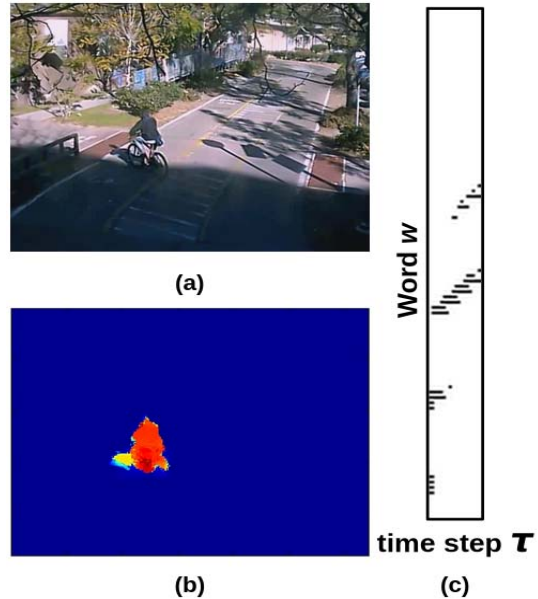

Fig. 4. (a) Example frame from a sequence belonging to the training video library. (b) Visualization of the optical flow magnitude for the shown frame. (c) Document generated from the sequence. The vertical axis corresponds to the word, which in turn corresponds to spatial location of a cube and the observed direction of motion. The horizontal axis corresponds to time steps. The document is binary valued with the black regions signifying activations. (Best viewed in color.)

videos recorded across ten camera views on the UCSB campus [35], [39]. The resolution of the videos is 320×240 and they are recorded at the rate of 24 frames/s. Note that this does not constrain the dimensions of test videos. The library videos capture scenes of pedestrians and bicyclists on campus bike-paths from various viewpoints. There are a total of 291 object tracks in the library. Example frames from the library are shown in Fig. 3. Human-generated annotations corresponding to individual objects are stored and indexed. To increase the diversity of motion patterns in the data set, we have generated horizontally and vertically flipped versions of library videos.

2) Video Document Generation: We divide the training videos into small nonoverlapping spatiotemporal cubes and compute dense optical flow across frames [34]. For each spatiotemporal cube, optical flow vectors are averaged over a time step and those exceeding a specified magnitude are binned into four directions (top, left, bottom, and right). The binning is performed as a soft decision where an optical flow vector can belong to two directions (e.g., left and top), the contribution being directly proportional to how close the vector is to these directions. The votes for each of the optical flow

vectors are summed up and thresholded. This generates a 4-b binary motion code for each cube. For our experiments, we have set the spatial size of cube to 20×20 and the temporal step size to four frames. The spatial locations and the motion code of the cubes are flattened to a single column vector. Each of the binary codes in the column vector is termed as words with them being denoted by the variable $w \in [0, W]$. W is the number of spatiotemporal cubes in a time step multiplied by the number of quantized directions. The value of W is derived as

$$W = \frac{I_X * I_Y * m}{c_X * c_Y} \tag{1}$$

where I_X and I_Y are the video width and height, c_X and c_Y are the spatiotemporal cube width and height, and m is the number of binary bits in the motion code. For our experiments, $W = 768$. We tried out different values for these design parameters and got the best performance for the values specified before. The horizontal axis represents time steps in the video, indexed by $t \in [0, T]$. An example document is shown in Fig. 4. Design of the video document is meant to capture spatial location and directions of object motions from training videos.

3) Motion and Track Indexing: To enable search and retrieval of motion patterns from training videos, we divide the documents along the temporal dimension into fragments. We choose a parameter T_f that denotes the document fragment length. This is the temporal duration of the basic retrievable segment of a library video that will be chosen and combined to represent a query video. A fragment is, hence, a contiguous subset of T_f columns from a video document. In our
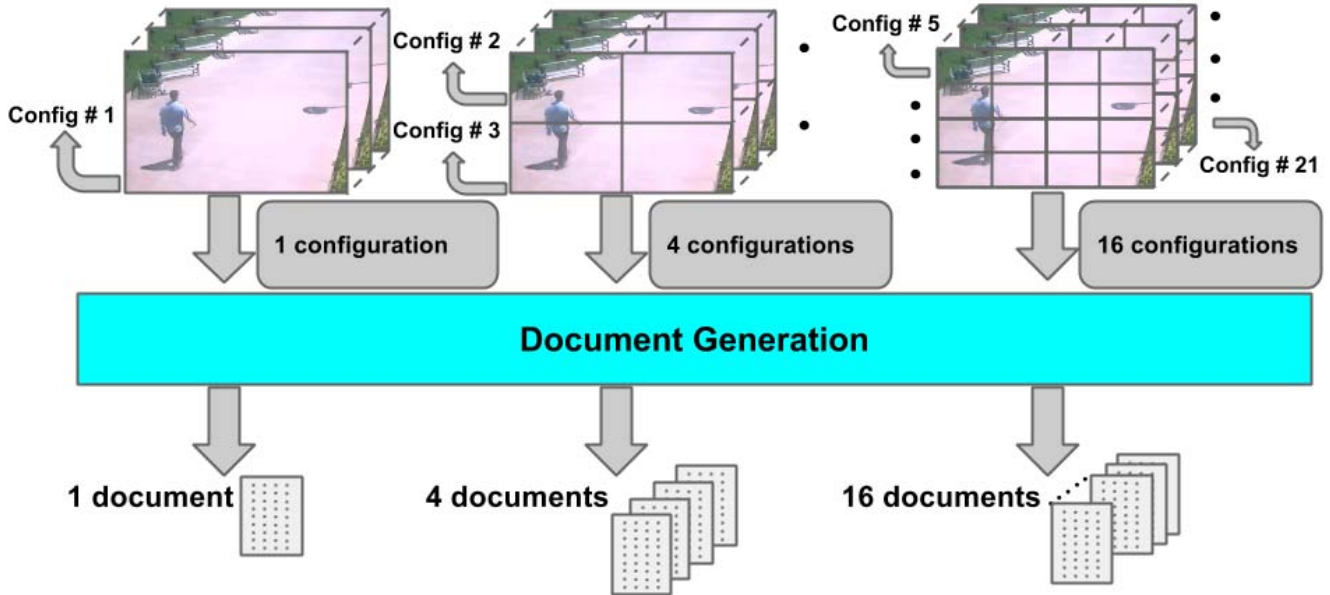
Fig. 5. Query video gets processed as 21 different spatiotemporal volume configurations. Each configuration gets processed into a document individually. (Best viewed in color.)

experiments, we have fixed T_f to 8. Each video fragment can be represented as a set of activated (w, τ) pairs, where $\tau \in [0, T_f)$ is the time relative to the start of the fragment. Each overlapping segment of a document with duration T_f is indexed as an individual fragment. During training, the library data are stored and indexed across five database tables.

1) *Fragment Forward Index:* This table contains a row for each fragment, mapping from a fragment name to its set of (w, τ) activations.
2) *Fragment Inverse Index:* This table contains a row for each (w, τ) pair, mapping onto the fragment names in which that pair appears.
3) *Flow Fields:* This table contains the optical flow magnitude for each time step in every document. These will be used later for warping.
4) *Track Forward Index:* This table contains a row for each unique track id present in the human-generated annotations, mapping onto a bounding box for each frame where the corresponding object is present.
5) *Track Inverse Index:* This table contains a row for each fragment, mapping onto the set of track ids annotated during that fragment's duration.

C. Online Video Queries

With offline library creation steps complete, the system is ready to provide tracks for a new unseen input video. Keeping with the search and retrieval metaphor, an input video is called a query.

1) Multiscale Video Document Generation: In order to be able to match motion patterns at multiple scales and spatial locations from the training video library, we generate documents for different configurations of the input video. The configurations are illustrated in Fig. 5. The first configuration has the video processed at the original scale.

The next four configurations have the video spatially divided into four quadrants. The quadrants are individually processed to create one document each. Additional 16 configurations are generated by spatially dividing the video into 16 parts of identical sizes and each part generating a document. In total, for each video, we generate 21 documents. The spatial dimensions of the spatiotemporal cubes used during document generation are modulated with the size of the video configuration such that the number of words W is constant across configurations. The above method enables the representation of motion patterns in query videos at different spatial locations and scales. When retrieving matches for query videos, we compute matches for all the 21 configurations and pool the results for further stages of annotation transfer and warping, as described in Section III-C3. This enhanced flexibility leads to a reduction in size of the training video library required to represent arbitrary object motion in query videos. We then divide the documents of the query video into fragments, as described in Section III-B2.

2) Library Search and Composition: Consider a fragment of one of the query video documents

$$f_q = (w, \tau) : w \in [0, W), \quad \tau \in [0, T_f). \quad (2)$$

We wish to find a set of result fragments from our library, $\mathcal{F}_r$, which composed together approximate the query fragment

$$\mathcal{F}_r = \arg\max_{\mathcal{F}'_r} \sum_w \sum_\tau \min(R_{f_q}(w, \tau), R_{f_R}(w, \tau)) \quad (3)$$

where

$$f_R = \bigcup_{f_r \in \mathcal{F}'_r} f_r \quad (4)$$

$$R_f(w, \tau) = \begin{cases} \dfrac{1}{|f|}, & \text{if } (w, \tau) \in f \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Algorithm 1 Greedy Composition of Library Fragments

Input:

Query fragment f_q

Fragment forward index I_f

Fragment inverse index I_i

Stopping criteria ρ

Output:

Result fragment set $\mathcal{F}_r$

1: $U \leftarrow f_q$
2: $U_0 \leftarrow |U|$
3: $\mathcal{F}_r \leftarrow \{\}$
4: $f_R \leftarrow \{\}$
5: $h \leftarrow 0$
6: **while** $|U| > \rho \, U_0$ **do**
7:      $X \leftarrow \bigcup_{(w,\tau) \in U} I_i[(w, \tau)]$
8:      $y \leftarrow \{\}$
9:      **for** $x \in X$ **do**
10:          $f_c \leftarrow f_R \cup I_f[x]$
11:          $h \leftarrow \frac{|f_q \cap f_c|}{\max(|f_q|, |f_c|)}$
12:          $y \leftarrow y \cup (h, x)$
13:      **end for**
14:      $h_m, x_m \leftarrow \max(y)$
15:      $f_R \leftarrow f_R \cup I_f[x_m]$
16:      $U \leftarrow f_q \backslash f_R$
17:      $\mathcal{F}_r \leftarrow \mathcal{F}_r \cup x_m$
18: **end while**

Here, f_R is the union of all the selected result documents and $R_f(w, \tau)$ is a function that represents a set f as a uniformly weighted discrete probability distribution whose support is the (w, τ) pairs in f. As such, we are searching for the set of library fragments where probability distribution for their union has a maximal histogram intersection with probability distribution for the query fragment. This can be rewritten as

$$\mathcal{F}_r = \arg\max_{\mathcal{F}_r'} \frac{|f_q \cap f_R|}{\max(|f_q|, |f_R|)}. \tag{6}$$

Choosing the library fragments to include in the result set $\mathcal{F}_r$ is very similar to the maximum set coverage problem, which is nondeterministic polynomial-time hard [13]. We approach the selection of $\mathcal{F}_r$ using a greedy algorithm, which at each step adds a new fragment from the set of library fragments to the result set such that the resulting histogram intersection is maximized. The retrieval algorithm is summarized in Algorithm 1. In detail, a set of fragments from the library X, which share activations with the query fragment f_q, are retrieved using the fragment reverse index I_i. We then find the fragments within X, which together compose f_q in a greedy fashion. In the case where library videos are provided as queries, the algorithm will produce an exact match in the first iteration and generated tracks will be the same as ground truth. See Fig. 6 for an example of one of the library fragments retrieved for a query fragment.

The retrieval algorithm scales with multiple objects in the query video. Consider an example where we have
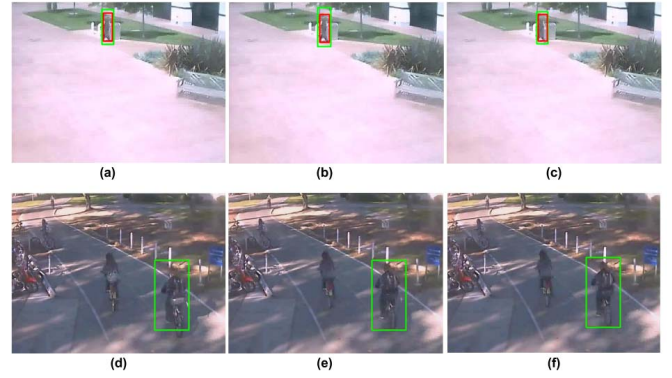

Fig. 6. Example of retrieval results for a query video sequence. (a)–(c) Frames from a query video and (d)–(f) frames from the top retrieved result among the library videos. The motion of the walking person in the query video in the top-left direction has been matched to the motion of the bicyclist. Note the difference in the spatial scales and locations of the objects in the query and result videos. The red boxes in (a)–(c) signify detected bounding boxes and green boxes signify ground truth. The green boxes in (d)–(f) show the human annotated bounding boxes stored with the library videos. (Best viewed in color.)

two objects moving in a frame, one moves to the left and the other to the right. Since the objects would occupy distinct spatial locations and would have different directions of motion, the activations get encoded in distinct locations of the corresponding document and consequently the fragments. This leads to two distinct motion patterns in the fragment. Each of the distinct patterns would result in retrieval results that compose these results independently. The design of the retrieval algorithm ensures that we get multiple composed fragments from the reference result with one corresponding to motion to the left and the other corresponding to motion to the right.

3) Annotation Transfer and Warping: The previous step resulted in $\mathcal{F}_R$, the set of library result fragments that together best approximate the query fragment. Looking up each of these fragment names in *track inverse index* gives the set of unique track ids occurring in the result fragments, and looking up each of these up in the *track forward index* gives a set of bounding boxes to be transferred to the query video. Finally, we retrieve optical flow magnitude fields for the result fragments from the *flow field* table. The indexes were previously defined in Section III-B3. Each fragment corresponds to T_f flow fields.

Fig. 7 shows the flow fields and annotations retrieved for the example shown in Fig. 6. Note that while motion of the bicyclist in the result fragment and the pedestrian in the query fragment are similar, the objects are of different sizes and are in different locations in the image frame. We cannot simply copy the bounding boxes from one to the other. Instead, the flow fields can be used to warp retrieved bounding boxes to better match the query.

It is not necessary to obtain a dense warping field from the result to the query; only the bounding box needs to be adjusted. The system seeks a bounding box on the query flow field that is similar to the human-provided bounding box on the result flow field. This includes both the size and placement of the box, as well as the flow it contains. Bounding boxes are defined by
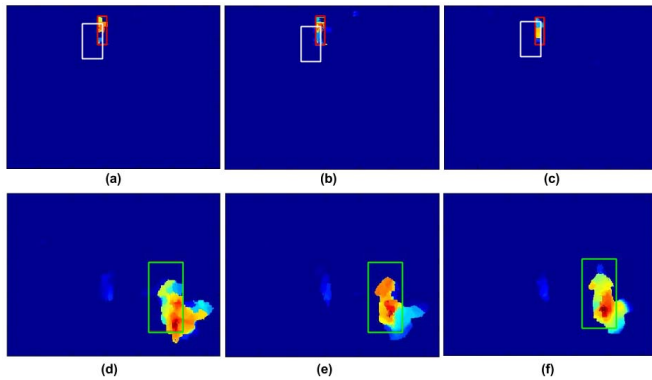
Fig. 7. (a)–(f) Optical flow fields for frames presented in Fig. 6. Query frame flow fields in (a)–(c) have the directly transferred white bounding boxes and the results of the warping method (described in Section III-C3) are drawn using the red bounding boxes. The green boxes in (d)–(f) represent manually annotated bounding boxes from the retrieval results. (Best viewed in color.)

their left, right, top, and bottom edge positions and the system iteratively updates each edge of the query bounding box in turn to improve quality of the match. For a query bounding box b_q and a result bounding box b_r with left edge values l_q and l_r, respectively, the update for l_q is

$$l_q = \arg\max_{l'_q} \left[\sum_{i=0}^{N-1} \min(H_{b_r}(i), H_{b'_q}(i)) \right] e^{\frac{-(l_r - l'_q)^2}{2\alpha}} \qquad (7)$$

where H_b is a N-bin normalized histogram of the flow magnitudes inside the bounding box b and α is a penalty factor. As such, the update seeks a new query edge position which maximizes the histogram intersection between the histograms of the flows in b_q and b_r and exacts a penalty for deviating too far from the result bounding box. The second part of the update criterion ensures that the query bounding box does not collapse onto a subregion of the query frame's optical flow.

The right, top, and bottom edges proceed similarly. The warping scheme does not put a rigid constraint on the size of the final bounding box and allows adaptation to optical flow statistics of the local neighborhood. Fig. 7 shows an example result of warping bounding boxes. In our experiments, we randomly permute the order of the left, right, top, and bottom edges and obtain a batch of updates to eliminate bias that the order of edges might introduce. We have observed that the values of the edges reliably converge within ten batches across multiple test matches. We have included a sensitivity analysis for α in Section IV-E.

Due to the overlapping nature of fragments, a frame can belong to multiple fragments. This leads to multiple bounding boxes being retrieved for a given motion pattern in a frame. To choose the best warped bounding box, we apply a nonmaximal suppression rule to eliminate suboptimal boxes. Bounding boxes are scored on the density of the optical flow being covered.

The chosen detection bounding boxes are associated together into object tracks using the Hungarian algorithm [24], [25] to solve an assignment problem where the association costs are modeled by a combination of geometric distance between bounding box centers and color

histogram distance. In detail, the association costs between bounding boxes b_i^n and b_j^{n+1} in frames n and $n + 1$ are modeled as

$$J_{ij}^{n,n+1} = d_{\text{hist}}\left(H_{b_i^n}^{\text{hsv}}, H_{b_j^{n+1}}^{\text{hsv}}\right) + \beta \|c_{b_i^n} - c_{b_j^{n+1}}\|_2 \qquad (8)$$

where H_b^{hsv} is the HSV color histogram of the image pixels lying within the bounding box b, $d_{\text{hist}}(.,.)$ is the histogram intersection distance, β is a weight parameter, and c_b is the center location of the bounding box b. The color histograms are constructed by jointly binning hue and saturation values. H and S channels are quantized into ten and five equally spaced bins respectively. The parameter β is fixed to 2.5 in our experiments, as due to poor image quality in our query videos, color information can be unreliable and provides only coarse discriminative information for association.

Once tracks are generated from the above step, we perform postprocessing in the form of a moving average filter with a window width of ± 2 frames. We perform this step to improve temporal coherence of the generated bounding boxes. The averaging operation is carried out in the center location and scale of the bounding boxes independently.

IV. EXPERIMENTS

A. Data Sets

We have focused our experiments on surveillance videos. As the proposed approach is designed to be effective for low-quality and low-resolution videos, we have collected an appropriate data set with 15 sequences. We call it the UCSB-Courtyard data set. These video clips have been recorded using Cisco WVC2300 wireless ip network cameras overlooking a busy pedestrian crossing from five different viewpoints. Each sequence contains on average 150 frames with pedestrians on a busy courtyard in an uncontrolled setting. The number of pedestrians varies from 1 to 4. The tracking targets undergo complex appearance changes due to shadows, occlusions, and compression artifacts. The *Browse2*, *WalkByShop1front*, *ShopAssistant1front*, *TwoEnterShop2cor*, *OneShopOneWait2cor*, and *OneLeaveShop1cor* sequences from the CAVIAR [9] data set are used for comparisons as well. These data sets are used to measure single-object tracking performance. The proposed method is also capable of detecting multiple moving objects in a scene. To compare and benchmark with respect to other multiple object trackers, we have chosen the *S2L2* sequence of PETS2009 [6].

As described earlier, we have composed the library videos from a data set that covers bike paths on a university campus. The scenes captured on this data set are distinct from test data sets. We demonstrate that with a small library of videos, we can apply learned motion patterns from one data set onto an entirely different data set.

B. Evaluation Metrics

To perform quantitative comparison of object tracking, we use the standard metrics of Pascal visual object challenge (VOC) detection score [7] and center location error (CLE) [38].
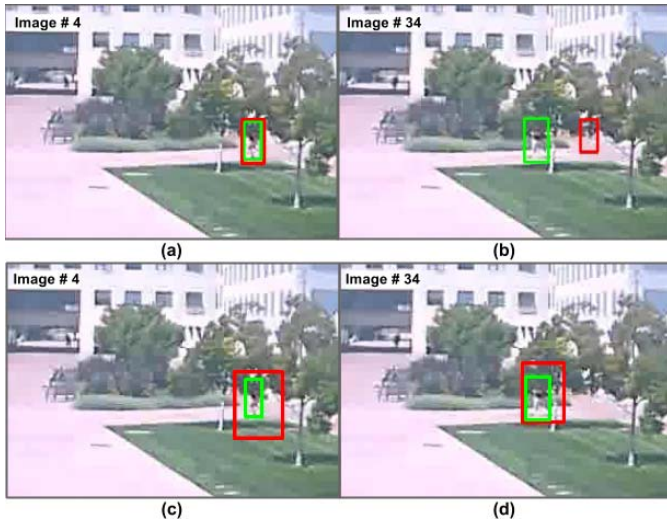
Fig. 8. (a) and (b) Results for appearance-based tracker reproduced from Fig. 1 for comparison. (c) and (d) Results for the ST on frames from the Courtyard data set. The ST has ignored scene clutter and continues to track the target across frames. Tracker results and ground truth boxes are marked in red and green, respectively. (Best viewed in color.)

VOC score measures the quality of overlap between detected and ground-truth bounding boxes. For comparison of VOC scores of the competing methods in a test data set, we average the scores over frames in a sequence, and then over sequences to get the final score to generate a mean VOC score. CLE measures the Euclidean distance between the center of the detected bounding box and that of the ground-truth bounding box. CLE quantifies the localization ability of an object tracker. Similar to mean VOC score, we calculate mean CLE score.

In addition to the above single object tracking metrics, we have also used CLEAR metrics [2] for comparison of algorithm performance with multiple object trackers. Multiple Object Tracking Precision (MOTP) scores measures the ability to detect precise object locations whereas Multiple Object Tracking Accuracy (MOTA) measures the capability of trackers to maintain consistent object configurations as targets move around in the scene.

C. Comparison With State-of-the-Art

In order to demonstrate the advantages of the proposed approach over those of more conventional appearance-based approaches, we have chosen six state-of-the-art methods for comparison.

1) *Visual Tracking Decomposition [19]:* This method combines multiple-appearance-based observation model and motion model trackers using sparse principle component analysis and an interactive Markov chain Monte Carlo framework. An initial bounding box of the target is required for tracking.

2) *Struck Tracking [12]:* This adaptive method formulates the problem of choosing good training examples for online training of target appearance as a structured support vector machine. An initialization of the target position is required for tracking.

TABLE I

COMPARATIVE TABLE OF MEAN VOC SCORES FOR DATA SETS ACROSS TRACKING METHODS. A HIGHER VALUE REFLECTS SUPERIOR TRACKING RESULTS

Algorithm	Courtyard	CAVIAR
Struck Tracker*[12]	0.2952	0.1338
VTD*[19]	0.2593	0.0858
ACT*[5]	0.3163	0.3291
IIT*[40]	0.2949	0.0743
BGS[17]	0.3803	0.4160
CMT*[26]	0.3114	0.1661
Search Tracker	**0.4539**	**0.6075**

The Algorithms marked with ∗ require manual initialization.

TABLE II

COMPARATIVE TABLE OF MEAN CLE SCORES IN PIXELS FOR DATA SETS ACROSS TRACKING METHODS. A LOWER VALUE REFLECTS SUPERIOR TRACKING RESULTS

Algorithm	Courtyard	CAVIAR
Struck Tracker*[12]	37.41	75.38
VTD*[19]	73.57	89.60
ACT*[5]	28.54	42.66
IIT*[40]	24.32	82.52
BGS[17]	49.13	38.39
CMT*[26]	59.54	81.66
Search Tracker	**21.06**	**27.46**

The Algorithms marked with ∗ require manual initialization.

3) *Adaptive Color Tracking [5]:* This real-time tracking method incorporates sophisticated color features to provide invariant representation in the illumination space. An initial bounding box of the target is required for tracking.

4) *Initialization-Insensitive Tracking [40]:* This approach utilizes motion saliency of local features to accurately track objects in an adaptive manner with inaccurate initializations. Target position initialization is required here as well.

5) *Consensus-Based Tracking and Matching of Keypoints for Object Tracking [26]:* This method tracks feature points across frames to estimate target location in current frame. Target position initialization is a requirement.

6) *Background Subtraction-Based Tracking [17]:* This method segments out moving objects in a scene from the background and applies a Kalman filter over bounding box estimates.

The results for the competing methods have been generated using codes provided by Danelljan *et al.* [5], Hare *et al.* [12], KaewTraKulPong and Bowden [17], Kwon and Lee [19], Nebehay and Pflugfelder [26], and Yi *et al.* [40], with parameters set to the default values suggested by the provided documentation.

In Fig. 1, we presented a case where scene clutter and compression artifacts can cause appearance-based trackers to fail. The ST can overcome such issues since quantized long-term motion patterns are robust to the presence of scene clutter and occlusions. Results for the ST on the same sequence are shown in Fig. 8. The ST is also robust to abrupt appearance changes due to shadows, compression artifacts, and changing illumination because of the relative invariance
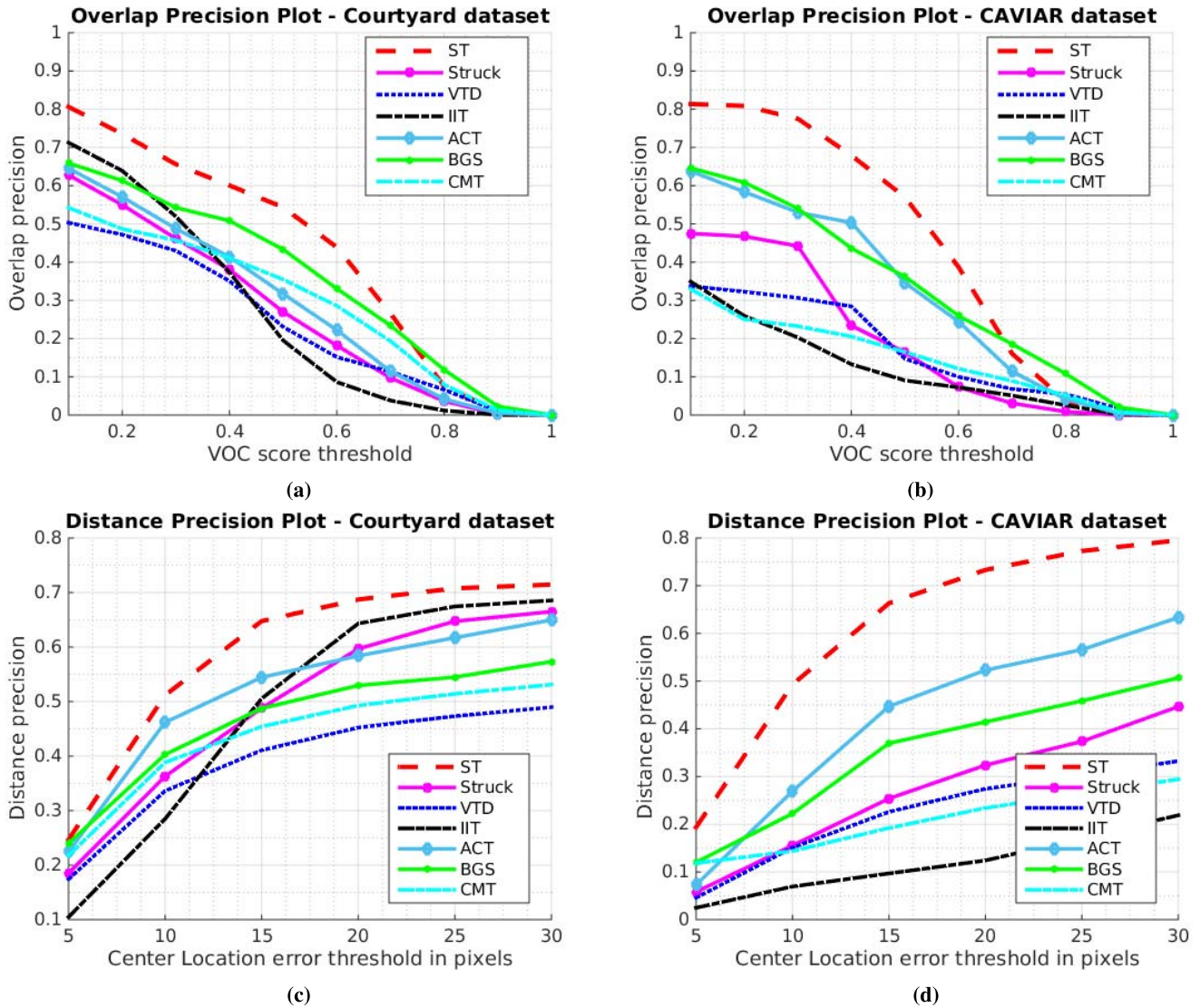
Fig. 9. Comparative distance and overlap precision performance plots for the ST and competing algorithms on the Courtyard and CAVIAR data sets.

TABLE III

COMPARATIVE TABLE OF MEAN OVERLAP PRECISION FOR DATA SETS
ACROSS TRACKING METHODS. THE OVERLAP THRESHOLD
IS SET TO 0.5. A HIGHER VALUE REFLECTS SUPERIOR
TRACKING RESULTS

Algorithm	Courtyard	CAVIAR
Struck Tracker*[12]	0.2698	0.1645
VTD*[19]	0.2306	0.1473
ACT*[5]	0.3172	0.3459
IIT*[40]	0.1951	0.0906
BGS[17]	0.4330	0.3621
CMT*[26]	0.3124	0.1649
Search Tracker	**0.5433**	**0.5700**

The Algorithms marked with ∗ require
manual initialization.

TABLE IV

COMPARATIVE TABLE OF MEAN DISTANCE PRECISION FOR DATA SETS
ACROSS TRACKING METHODS. THE DISTANCE THRESHOLD
IS SET TO 20 PIXELS. A HIGHER VALUE REFLECTS
SUPERIOR TRACKING RESULTS

Algorithm	Courtyard	CAVIAR
Struck Tracker*[12]	0.5968	0.3238
VTD*[19]	0.4523	0.2743
ACT*[5]	0.5839	0.5230
IIT*[40]	0.6432	0.1240
BGS[17]	0.5295	0.4145
CMT*[26]	0.5197	0.2340
Search Tracker	**0.6873**	**0.7331**

The Algorithms marked with ∗ require
manual initialization.

of long-term motion patterns, whereas appearance-based trackers frequently fail in such sequences. These issues are very important to address as they are commonplace in real-world scenarios.

Tables I and II show the comparison of mean VOC and mean CLE scores across different data sets between the proposed method and competing methods. Tables III and IV report comparative results on mean overlap precision and

mean distance precision across data sets and methods. The distance and overlap thresholds are set to 20 pixels and 0.5, respectively. Fig. 9 presents distance and overlap precision scores for different values of VOC scores and CLE thresholds. The ST consistently outperforms all other competing algorithms by a wide margin.

As we can see, the ST is competitive with respect to the appearance-based methods. It is important to note that we do not depend on manually provided initial bounding boxes or object detectors for the training videos. This gives us a strong advantage when the manual initialization or good object detectors are not available especially in test data sets suffering from poor image quality. The ST outperforms competing methods by a large margin in the Courtyard data set. We are able to get this performance from the ST without any manual initialization. CAVIAR has indoor sequences set in a shopping mall with a comparatively low image quality and more scene clutter. Therefore, leveraging motion patterns helps us outperform all the other algorithms on CAVIAR. Example result frames are presented in Fig. 10. These frames illustrate resilience of our algorithm to scene clutter, illumination changes, and occlusions. In addition, the aforementioned image quality issues often cause background subtraction-based tracking (BGS) methods to fail on both the CAVIAR and the Courtyard sequences. To contrast against the usage of optical flow and feature tracking methods, we have provided comparisons with the consensus-based tracking and matching of keypoints for object tracking (CMT) tracker. Due to poor image quality of test videos, consistent tracking of object feature points across multiple points is a difficult problem and hence leads to a comparatively weaker tracker performance. Since the ST utilizes aggregated optical flow information across multiple frames at the same time, the tracker is robust to such conditions.

In order to compare the performance with respect to tracking multiple objects, we provide the bounding boxes generated by the Search Tracker on the PETS 2009 S2L2 sequence to [23]. [23] combines the provided detections into object tracks using an energy minimization framework. We compute the MOTA and MOTP scores generated for these tracks and compare them with the state-of-the-art methods in Table V. Our method is comparable in performance with other multiobject trackers. A point to note is that the competing methods use external sources for object bounding boxes.

D. Performance Analysis With Varying Library Sizes

We investigate the effect of different library sizes on the proposed method's tracking performance. We randomly chose $\gamma = \{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ fraction of the library videos and generate sub-libraries. We then run the search and retrieval algorithm with one of these sublibraries at a time and plot the overlap precision and the distance precision scores on the Courtyard data set for the different values of γ in Fig. 11(a). As can be seen from the plots, the ST's performance scales with the size of the associated annotated video library. Since we apply data augmentation techniques in the form of vertical

Algorithm	MOTA	MOTP
Milan et al.* [22]	56.9	**59.4**
JPDA$_{100}$ * [28]	**59.3**	58.27
Search Tracker	46.4	55.8

Note that both metrics are normalized such that scores of 100 for both MOTA and MOTP correspond to perfect alignment of tracker generated bounding boxes to ground truth and no identity switches amongst tracks. Algorithms marked with * require external object detectors.

and horizontal flipping of library videos and also generate multiscale query video representations, the proposed method's performance does not reduce by a large margin due to reduction in library sizes.

E. Analysis on Annotation Warping

In the annotation warping stage, we control the flexibility that a transferred bounding box has in fitting optical flow characteristics of the query video frame, through the penalty term α from (7). We found the optimal value of α to be 2000 for our experiments. To investigate the sensitivity of the proposed method for different values of α, we execute the proposed tracker on the Courtyard data set and measure the overlap precision and distance precision at VOC score thresholds of 0.5 and 20 pixels, respectively. The tracking performance of the ST is shown in Fig. 11(b). Low values of α restrict the flexibility of the transferred bounding box to adapt the test sequence's optical flow characteristics, while higher values can lead to bounding boxes collapsing onto regions of high optical flow magnitude.

F. Computational Cost

Our experiments were carried out on a single-core 3.5-GHz workstation using MATLAB. The query stage and the bounding box composition steps take between 4 and 25 s for each frame, depending on the number of moving objects in the scene. The computational cost of the ST is distributed among the query multiscale fragment computation stage, the library search and composition stage, and the annotation transfer and warping stage. The time required per frame for fragment generation is 53 ms, the library search stage needs 3.7 s, and the annotation transfer stage requires 9.3 s on average for the Courtyard data set.

The cost of fragment generation is independent of the content in query videos. Annotation transfer and warping requires the largest amount of computation among all the stages. Since a frame can be a member of multiple query fragments, the large number of matched annotations and the accompanying warping procedure adds to the computational cost. Annotation warping can be made faster by a parallelized implementation for warping of retrieved candidate bounding boxes. The optical flow method in [34] provided the most
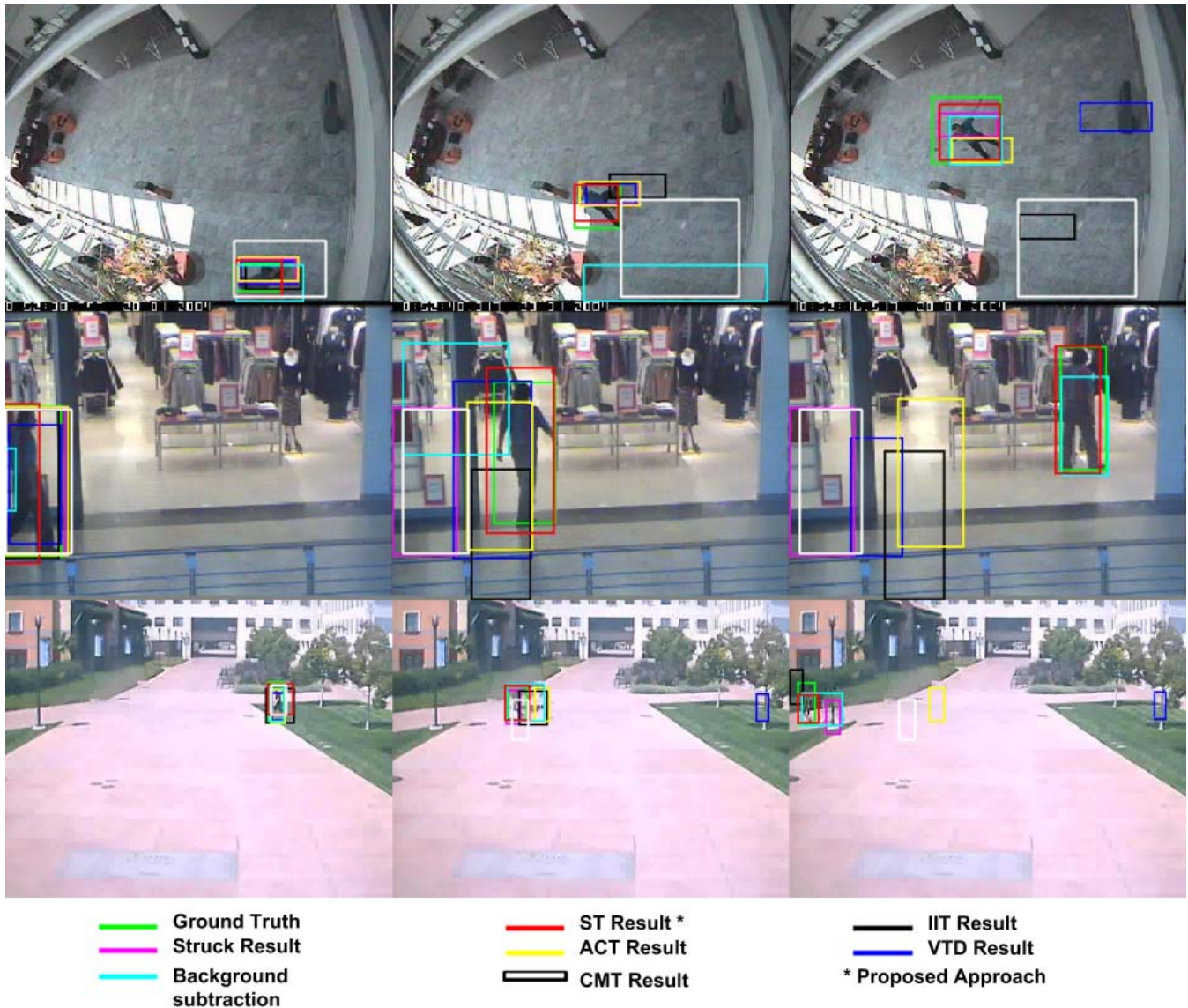
Fig. 10.   Comparison of our approach with state-of-the-art trackers on CAVIAR and Courtyard data sets. The first (top) row shows images where the target undergoes illumination and shape variations. In the second row, the target passes through a cluttered scene. The target in the third row undergoes compression artifacts and an occlusion. The proposed tracker is able to track the targets and it adapts bounding box scale to the target size, whereas the competing trackers get distracted by scene clutter, have fixed bounding box scales, and fail when the target appearance changes or undergoes occlusions. (Best viewed in color.)

accurate results, but the method is computationally expensive and this adds to the cost of the ST.

V. Discussion

There are a few limitations to the proposed method. The ST is designed to work with stationary cameras and will not be directly applicable to data from pan–tilt–zoom and mobile device cameras. There may be cases where the motion present in the test video cannot be modeled by the training library database, which can be overcome by adding more video clips to the library. Diversity can also be induced by generating translated and rotated versions of preexisting library videos. In addition, with state-of-the-art trackers becoming more efficient and robust, we could combine automated tracker outputs instead of depending on human-generated

annotations to create cheaper large-scale video libraries and, consequently, lead to improved object tracking. We also expect that this method of directly transferring knowledge available on one annotated data set to a different data set to be applicable to other problems like action recognition, activity analysis, and other tasks that can be analyzed through motion patterns.

The ST has also limitations with respect to modeling target motion in crowded sequences. In sequences where a large number of targets occlude each other, the optical flow signatures are not discriminative enough to find a good match from the library data set. In some cases, very small objects in scenes do not generate strong optical flow fields, and hence, encoding of motion becomes challenging. The ST is best suited for tracking
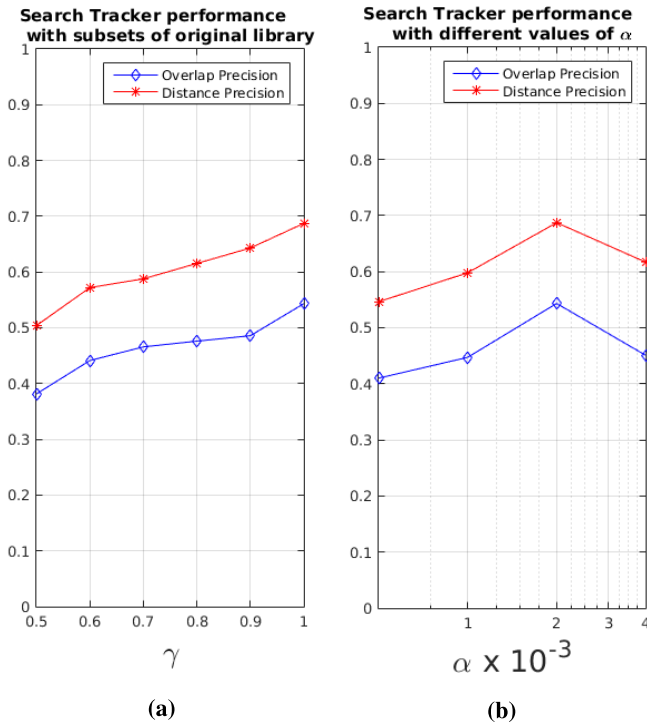
Fig. 11. (a) Overlap precision and distance precision values for the proposed method on the Courtyard data set with γ fraction of the entire video library used for the retrieval database. (b) Overlap precision and distance precision values for the proposed method on the Courtyard data set for different values of the parameter α. The VOC score threshold and CLE thresholds were set as 0.5 and 20 pixels, respectively, for both plots and Courtyard was used as the test data set to generate both plots.

a fewer number of objects in cluttered and challenging scenarios.

VI. CONCLUSION

In the proposed method, the transferred annotations are warped on each frame from the query video. The warping algorithm could be made more robust and efficient by considering optical flow characteristics of adjacent frames, resulting in smoother tracks.

The paradigm of learning motion patterns and behaviors from an annotated library of past videos can be extended to several novel surveillance scenarios. Consider a surveillance network where we have annotations for videos from a subset of the connected cameras. With the remaining cameras or in the event of adding a new camera, we could directly start leveraging the past motion pattern knowledge mined from the annotated data set. The ST could also be used in an active learning framework where imperfect appearance-based trackers and detectors are used as teacher algorithms to create a seed library. The ST as the student algorithm tracks objects in conditions that are difficult for appearance-based trackers using the library. The library expands continuously, both from the past outputs of the ST and the appearance-based tracker, which would lead to an improvement in ST performance. The basic idea of similarity search of motion patterns could be explored for applications in action recognition, object retrieval, and object reidentification from videos.

We have presented a novel approach to tracking that uses human annotations to directly drive an automated tracking system. We generate documents from videos that represent motion patterns. These documents are used to retrieve videos with similar motion characteristics and associated annotations are transferred and warped to the query video. This system avoids the requirement of object detectors and outperforms the state-of-the-art appearance-based trackers on the in-the-wild surveillance data sets, which has been demonstrated in the experiments.

REFERENCES

[1] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.
[2] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *J. Image Video Process.*, vol. 2008, p. 246309, May 2008.
[3] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Proc. 4th IEEE Workshop Appl. Comput. Vis. (WACV)*, Oct. 1998, pp. 214–219.
[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.
[5] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1090–1097.
[6] A. Ellis, A. Shahrokni, and J. M. Ferryman, "PETS2009 and winter-PETS 2009 results: A combined evaluation," in *Proc. 12th IEEE Int. Workshop Perform. Eval. Tracking Surveill. (PETS-Winter)*, Dec. 2009, pp. 1–8.
[7] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[8] J. M. Ferryman and A. Ellis, "PETS2010: Dataset and challenge," in *Proc. AVSS*, Aug./Sep. 2010, pp. 143–150.
[9] *The CAVIAR Website*, accessed on May 19, 2016. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/
[10] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. Brit. Mach. Vis. Conf.*, 2006, pp. 1–10.
[11] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," in *Proc. 10th IEEE Int. Workshop Perform. Eval. Tracking Surveill. (PETS)*, Sep. 2007, pp. 1–7.
[12] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. ICCV*, Nov. 2011, pp. 263–270.
[13] D. S. Hochbaum and W. Maass, "Approximation schemes for covering and packing problems in image processing and VLSI," *J. ACM*, vol. 32, no. 1, pp. 130–136, Jan. 1985.
[14] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1296–1311, Oct. 2003.
[15] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proc. ICCV*, Oct. 2007, pp. 1–8.
[16] F. Jurie and M. Dhome, "Real time robust template matching," in *Proc. Brit. Mach. Vis. Conf.*, 2002, pp. 123–131.
[17] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*, P. Remagnino, G. A. Jones, N. Paragios, and C. S. Regazzoni, Eds. New York, NY, USA: Springer, 2002, pp. 135–144.
[18] S. Karthikeyan, V. Jagadeesh, R. Shenoy, M. Ecksteinz, and B. S. Manjunath, "From where and how to what we see," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 625–632.
[19] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. CVPR*, Jun. 2010, pp. 1269–1276.
[20] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Van Den Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, pp. 58:1–58:48, Oct. 2013.
[21] S. Manen, J. Kwon, M. Guillaumin, and L. Van Gool, "Appearances can be deceiving: Learning visual tracking from few trajectory annotations," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 157–172.
[22] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[23] A. Milan, K. Schindler, and S. Roth, "Detection- and trajectory-level exclusion in multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3682–3689.

[24] M. L. Miller, H. S. Stone, and I. J. Cox, "Optimizing Murty's ranked assignment method," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 3, pp. 851–862, Jul. 1997.

[25] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[26] G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2784–2791.

[27] D. P. Papadopoulos, A. D. F. Clarke, F. Keller, and V. Ferrari, "Training object class detectors from eye tracking data," in *Computer Vision—ECCV*. Zurich, Switzerland: Springer, 2014, pp. 361–376.

[28] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proc. ICCV*, Dec. 2015, pp. 3047–3055.

[29] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert, "Data-driven crowd analysis in videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2011, pp. 1235–1242.

[30] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof, "Online multi-class LPBoost," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3570–3577.

[31] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 300–312, Feb. 2007.

[32] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[33] D. Song and D. Tao, "Biologically inspired feature manifold for scene classification," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 174–184, Jan. 2010.

[34] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2432–2439.

[35] S. Sunderrajan, J. Xu, and B. S. Manjunath, "Context-aware graph modeling for object search and retrieval in a wide area camera network," in *Proc. 7th Int. Conf. Distrib. Smart Cameras (ICDSC)*, Oct./Nov. 2013, pp. 1–7.

[36] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, pp. I-511–I-518.

[37] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive object tracking based on an effective appearance filter," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1661–1667, Sep. 2007.

[38] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2411–2418.

[39] J. Xu *et al.*, "Graph-based topic-focused retrieval in distributed camera network," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2046–2057, Dec. 2013.

[40] K. M. Yi, H. Jeong, B. Heo, H. J. Chang, and J. Y. Choi, "Initialization-insensitive visual tracking through voting with salient local features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2912–2919.

[41] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006, Art. no. 13.

S. Karthikeyan (S'09) received the B.Tech. degree in electrical engineering from IIT Madras, Chennai, India, in 2008, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2010 and 2014, respectively.

He joined Synaptics, Inc., San Jose, CA, USA, in 2015, where he is currently an Algorithms Architect. His current research interests include computer vision and pattern recognition, in particular object detection, tracking, and visual attention modeling.

Carter De Leo was born in Mather Air Force Base, CA, USA, in 1986. He received the B.S. degree in electrical engineering from California Polytechnic State University, San Luis Obispo, CA, USA, in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2008 and 2013, respectively.

He is currently a Senior Software Engineer with Google, Mountain View, CA, USA. His current research interests include video summarization and search and retrieval.

Santhoshkumar Sunderrajan (S'13–M'14) received the B.E. degree from Anna University, Chennai, India, in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2009 and 2014, respectively.

He is currently a Senior Member of the Technical Staff with Pinger, Inc., San Jose, CA, USA. His graduate research work is focused on multiple camera tracking, object reidentification, and activity analysis. His current research interests include computer vision and large scale machine learning.

Dr. Sunderrajan's paper describing the tracking research received the Excellent Paper Award at the ACM/IEEE International Conference on Distributed Smart Cameras in 2013.

Archith John Bency (S'16) received the B.Tech. degree in electrical engineering from IIT Bombay, Mumbai, India, in 2009, and the M.S. degree in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2014, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering.

He was involved in multimedia frameworks for video conferencing and surveillance systems with Texas Instruments Inc., Bengaluru, India, from 2009 to 2012. He is also a Graduate Student Researcher with the Department of Electrical and Computer Engineering, University of California at Santa Barbara. His current research interests include video and image analysis for object detection and tracking, deep neural networks, and machine learning.

B. S. Manjunath (F'05) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1991.

He is currently a Professor of Electrical and Computer Engineering with the University of California at Santa Barbara, Santa Barbara, CA, USA, where he directs the Center for Multi-Modal Big Data Science and Healthcare. He has authored or co-authored about 300 peer-reviewed articles. He is a Co-Editor of the book entitled *Introduction to MPEG-7* (New York: Wiley, 2002). His current research interests include image processing, computer vision, biomedical image analysis, and large-scale multimedia databases.